



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/717,671	11/21/2003	Anuj Dua	10001618-3	9068

22879 7590 11/13/2006

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER

HUISMAN, DAVID J

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 11/13/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

10/717,671

Applicant(s)

DUA ET AL.

Examiner

David J. Huisman

Art Unit

2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 19 September 2006.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-15 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☒ Claim(s) 1-10 is/are allowed.
- 6) ☒ Claim(s) 11-15 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 21 November 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Claims 1-15 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment and Terminal Disclaimer as received on 9/19/2006.

Terminal Disclaimer

3. The terminal disclaimer filed on September 19, 2006, disclaiming the terminal portion of any patent granted on this application which would extend beyond the expiration date of U.S. Patent No. 6,678,817, has been reviewed and is accepted. The terminal disclaimer has been recorded.

Maintained Rejections

4. Applicant has failed to overcome the prior art rejections set forth in the previous Office Action. Consequently, these rejections are respectfully maintained by the examiner and are copied below for applicant's convenience.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person

Art Unit: 2183

having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claims 11, 12, and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kahle et al., U.S. Patent No. 5,732,235 (herein referred to as Kahle) and Kane et al., U.S. Patent No. 5,537,559 (herein referred to as Kane).

7. Referring to claim 11, Kahle has taught a multi-architecture computer system capable of implementing a native instruction set architecture (ISA) and an emulated ISA comprising:

a) a memory subsystem of a native ISA. See Fig.1 and column 2, lines 33-39.

b) a fetch engine of the native ISA, said fetch engine being electrically connected to the memory subsystem of the native ISA, wherein the fetch engine accesses the memory subsystem to retrieve a line of instructions from the memory subsystem. See Fig.1, and note that instructions are fetched into instruction queue 20. In addition, it is inherent that the native fetching unit and the memory in which native instructions are stored be electrically connected together; otherwise, it could not fetch from memory (note that the fetch engine could comprise the prefetch component 52 (Fig.2) and the buses the data cache to component 36). Finally, see column 5, lines 6-9, and note that every cycle, two instructions are fetched (i.e., the fetch bandwidth is two). These two instructions make up a line of instructions, where the line is associated with the fetch address applied to the memory in that cycle.

c) an engine of an emulated ISA, wherein the engine of the emulated ISA is electrically connected to the fetch engine and interfaces with the fetch engine using a handshake protocol, wherein the engine of the emulated ISA receives a line of instructions. See Fig.2 and note that the native fetch engine is connected to the emulation engine. In general, handshaking is the general communication between two components in order to complete a task. In Kahle, the fetch

Art Unit: 2183

engine fetches instructions and sends the guest instructions to the emulation engine 36. The emulation engine will then convert the guest instructions into native instructions and send them to the instruction queue portion of the fetch engine. See column 2, lines 57-67. So, the handshaking protocol comprises the fetch engine telling the emulation engine to take action on guest instructions and the emulation engine then tells the fetch engine that the action has been taken and that processing may continue on the converted guest instructions. And, recall that a line of instructions is received. See column 5, lines 6-9.

d) Kahle has not taught that the engine of the emulated ISA receives a fetch complete signal from the fetch engine. However, Kane has taught sending a fetch complete signal to an instruction buffer (Fig.4, component 480) in the form of an exception status signal. It should be noted that this status signal arrives simultaneously with the fetched instruction, since they go hand-in-hand. See column 11, lines 60-67, of Kane. Therefore, this signal is a signal separate from the line of instructions that signifies that the fetch is complete. This exception status signal allows for a simple solution for tracking address-exceptions and generating the exceptions at the appropriate point in time (i.e., immediately upon execution). See column 3, lines 57-65, of Kane. A person of ordinary skill in the art would have recognized that such exception signals would be applicable in Kahle's system since Kahle is concerned with address exceptions. From Fig.2 (component 54) of Kahle, it is seen that limit and attribute checks are made and exceptions are generated by the guest instructions. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to provide such a signal to the instruction buffer (Fig.2, component 50) in Kahle's emulation engine system which not only acts as a fetch complete signal, but also improves the efficiency of a system dealing with exceptions.

Art Unit: 2183

e) Kahle has not taught a fetch address queue that stores a fetch address for the line of instructions retrieved from the memory subsystem, wherein the fetch address queue is controlled by the fetch complete signal such that the fetch address is stored in the fetch address queue until the fetch complete signal is received. However, Kane has taught such a concept. See Fig.4, component 400, and column 11, lines 30-47. In this passage, Kane has disclosed that the reason fetch addresses are buffered (and postponed) is because operand (data) fetches have higher priority so that the operation of the CPU is not unnecessarily suspended by pending fetch requests. A person of ordinary skill in the art would have recognized that this concept is applicable to Kahle since emulation instructions and data are stored in the data cache (just as instructions and data are stored in the same cache within Kane's system). See Fig.1, column 2, lines 57-59, and column 3, lines 13-14, of Kahle. Since data and instructions are fetched from the data cache, then a fetch request and an operand request occurring at the same time would result in contention on the bus. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Kahle to include a fetch address queue as taught in Kane in order to store and postpone fetch requests to avoid unnecessary suspension of the CPU. Kane has further taught that the step of storing comprises storing the fetch address in the fetch address queue until the fetch complete signal is sent. It is disclosed, in column 12, lines 11-13, of Kane, that the fetch addresses are kept in the fetch address queue until the fetch request is performed. Consequently, the complete signal will signify that the fetch request has been performed (since it will arrive at the instruction buffer at the same time as the fetched instruction), and as a result, the corresponding fetch request will be removed from the queue.

Art Unit: 2183

8. Referring to claim 12, Kahle in view of Kane has taught a computer system as described in claim 11. Kahle has further taught that the engine of the emulated ISA requests the line of instructions and the fetch engine sends the line of instructions to the engine of the emulated ISA. See Fig.2, and note that when the emulation engine decodes a branch instruction (as shown in Fig.4 and Fig.6, for instance), the branch history table is accessed, which would in turn provide a target address fetch request (if the branch is predicted taken) to the prefetch mechanism for fetching a line of instructions. This line would then be sent from the memory subsystem to the emulation engine for execution.

9. Referring to claim 14, Kahle in view of Kane has taught a computer system as described in claim 11. Kahle has further taught a macroinstruction queue that stores the instructions that were retrieved by the fetch engine. See Fig 2, component 50.

10. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Kahle in view of Kane and further in view of Whitted III et al., U.S. Patent No. 5,515,521 (herein referred to as Whitted).

11. Referring to claim 13, Kahle in view of Kane has taught a computer system as described in claim 11. Kahle in view of Kane has not taught that if a pending fetch request is canceled due to a pipeline flush, then a pending fetch request is canceled and a fetch address queue is cleared. However, Whitted has taught this type of procedure. See column 9, lines 54-67. A person of ordinary skill in the art would have recognized that in branch situations, either the target address or the subsequent address would be fetched next. If the target address is to be fetched next (i.e. taken branch), then all of the subsequent pending requests that are on the non-taken path do not

Art Unit: 2183

need to be executed. Therefore, canceling the following pending requests that are unrelated to the branch along with clearing the fetch address queue would ensure that improper fetches are discarded and rollback is avoided (rollback being the term used to describe the corrections made to values that were incorrectly updated by instructions that should not have executed).

Therefore, in order to avoid fetching and executing improper instructions, it would have been obvious to one of ordinary skill in the art at the time of the invention to cancel a pending fetch request and clear the fetch address queue due to a pipeline flush.

12. Claim 15 is rejected under 35 U.S.C. 103(a) as being unpatentable over Kahle in view of Kane and further in view of Papworth et al., U.S. Patent No. 5,584,037 (herein referred to as Papworth).

13. Referring to claim 15, Kahle in view of Kane has taught a computer system as described in claim 14. Kahle in view of Kane has not taught the a speculative write pointer that prevents the macroinstruction queue from becoming oversubscribed by one or more pending fetch requests, wherein the speculative write pointer may be used to control the sending of a fetch request. However, Papworth has taught such the concept of using a speculative write pointer to track when a queue will be filled based on fetch requests. See column 2, lines 20-33. Clearly, once the queue is full, the queue will no longer accept data until some data is taken out. As a result, once the queue is full, it is inherent that fetch requests be controlled (halted), otherwise data could be incorrectly overwritten in the queue. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Kahle in view of Kane to

Art Unit: 2183

include a speculative write pointer for tracking queue fullness and controlling the sending of fetch requests accordingly.

Allowable Subject Matter

14. Claims 1-10 are allowed.

Response to Arguments

15. Applicant's arguments filed on September 19, 2006, have been fully considered but they are not persuasive.

16. Applicant argues the novelty/rejection of claims 11, 12, and 14, on page 7 of the remarks, in substance that:

"Kane nowhere identifies or describes a "fetch complete" signal or a signal that signifies that a fetch has been completed. The "fetch-exception status bits" convey data regarding what kind of exception is associated with each instruction byte, i.e., either a limit error, page fault, breakpoint, or no exception. (Kane, Col.11, ll: 16-22; Col. 12, ll. 1-3; Fig. 1; Col. 6, ll. 27-48). However, Kane neither states nor implies that the fetch-exception status bits signify "fetch complete." Indeed, the state of the fetch exception status bits in Kane is independent of the completion status of the fetch request, since the fetch request status bits for each fetch request are stored as data in the fetch address queue along with the corresponding fetch address until the fetch request is completed."

17. These arguments are not found persuasive for the following reasons:

a) The examiner asserts that applicant is reading the claims too narrowly. Applicant appears to be reading "fetch complete signal" as a signal which indicates that a fetch is finished. However, the claim does not specifically define the signal. The examiner's interpretation of "fetch complete signal" is some signal that is associated with a completed fetch. Kane's exception status signal is such a signal. The exception status signal is received when the fetch is completed (column 11, lines 60-67 of Kane), and consequently, the exception status signal is a "fetch complete signal"

Art Unit: 2183

18. Applicant argues the novelty/rejection of claims 11, 12, and 14, on page 7 of the remarks, in substance that:

"Although (as noted by the Examiner) the fetch status bits are provided simultaneously with the fetch data to the instruction buffer (Kane, Col. 11, ll. 64-67), Kane nowhere describes the fetch-exception status bits as "controlling" the fetch address queue."

19. These arguments are not found persuasive for the following reasons:

a) The removal of the fetch complete signal from the queue allows the entries to be moved to the next position (as is known in queues). Hence, the signal controls the queue in at least that aspect (by being removed, the bits allow the queue to advance the remaining data). Applicant has claimed how the queue is controlled. Applicant has merely claimed that the fetch address is stored until the signal is received.

20. Applicant argues the novelty/rejection of claims 11, 12, and 14, on pages 7-8 of the remarks, in substance that:

"The Examiner finds such motivation because a person of ordinary skill (i) would have recognized that the exception signals of Kane "would be applicable in Kahle's system since Kahle is concerned with address exceptions," and (ii) would have modified Kahle "to include a fetch address queue as taught in Kane to store and postpone fetch requests to avoid unnecessary suspension of the CPU." (Office action, at 12-13). Neither of these issues, however, is part of the problem addressed by Kahle. Kahle is concerned with reducing the cycle time required to process a sequence of emulated instructions, and it solves the problem by eliminating the need for a separate fetch request to decode a branch instruction in some instances. (Kahle, Col. 1, ll. 14 to Col. 2, ll. 4; Col. 4, ll. 28-40). Kahle expresses no concern regarding delays in fetching the instructions caused by incomplete fetches or bus suspensions."

21. These arguments are not found persuasive for the following reasons:

a) Just because a reference does not mention a concern does not mean it doesn't exist. There are many, many concerns in every system, and one cannot expect every single patent to address every single concern. Kahle is concerned primarily with minimizing the number of cycles

required to execute semantic routines. However, this doesn't mean Kahle is not concerned with other items. For instance, divide by zero cannot occur. One would expect that Kahle's system is concerned with that even if it is not explicitly disclosed. The examiner simply believes that one of ordinary skill would recognize concerns in Kahle and modify Kahle to address those concerns. In this case, Kahle does deal with exceptions (Fig.2, component 54). Hence, in a system that deals with exceptions, the examiner feels it would be obvious to include the exception signal of Kane which allows for tracking address exceptions to improve efficiency. Likewise, for the fetch address queue, since it avoids unnecessary suspension of the SPU, it would be obvious to modify Kahle to include such a queue to prevent such suspension.

Conclusion

22. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Art Unit: 2183

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (571) 272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

DJH
David J. Huisman
November 8, 2006


EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100